曲阜師範大學

本科生毕业论文（设计）

题　　目　基于真实对话日志的对话领域多分类的设计与实现

姓　　名　白振翰　学号　2019414212

院　　系　网络空间安全学院

专　　业　软件工程

指导教师　刘红娟　职称　副教授

2023 年　05 月 10 日

曲阜师范大学教务处制

目 录

摘要 1
Abstract 1
1 绪论 2
1.1 选题背景 2
1.2 对话领域多分类的挑战 3
1.2.1 数据集的匮乏 3
1.2.2 ASR 识别的不准确性 3
1.2.3 用户表述的不规范性 3
1.2.4 对话多领域的混合性 3
1.3 国内外研究现状 4
1.4 主要工作及意义 5
1.5 组织架构 6
2 相关技术与理论基础 7
2.1 Word2Vec 词向量的表示 7
2.2 预训练模型 8
2.2.1 Seq2Seq 8
2.2.2 注意力机制 9
2.2.3 Transformer 框架 10
2.2.4 BERT 及其变体 11
2.3 基于深度学习的分类方法 12
2.3.1 卷积神经网络 12
2.3.2 循环神经网络及其变体 13
2.4 本章小结 13
3 基于预训练模型的分类方法 14
3.1 问题描述 14
3.2 算法描述 14
3.3 基于 BERT 及其变体的基础模型 15
3.4 基于 ERNIE 的改进模型 16
3.4.1 基于 ERNIE 的 BiGRU 模型 17

3.4.2 基于 ERNIE 的 TextCNN 模型 17
3.4.3 基于 ERNIE 的 RCNN 模型 18
3.5 基于数据增强的改进模型 19
3.6 本章小结 19
4 实验分析与讨论 20
4.1 实验数据集 20
4.2 评估指标 21
4.3 实验准备 22
4.3.1 实验环境 22
4.3.2 预处理 23
4.4 实验设计 23
4.4.1 实验参数设置 24
4.5 实验结果对比分析 26
4.5.1 基于 BERT 及其变体的基础模型 26
4.5.2 基于 ERNIE 的改进模型 27
4.5.3 基于数据增强的改进模型 28
4.5.4 实验结果总结 29
4.6 本章小结 29
5 总结与展望 30
5.1 工作总结 30
5.2 未来展望 30
致谢 31
参考文献 32

图目录

图 1.1 任务导向型对话系统流程 2
图 1.2 预训练模型历史 5
图 2.1 Word2Vec 的两种实现方式[7] 7
图 2.2 Seq2Seq 架构 8
图 2.3 Attention 的模型结构[28] 9
图 2.4 单头和多头自注意力机制[13] 10
图 2.5 Transformer 框架[13] 11
图 2.6 BERT 的模型结构[14] 11
图 2.7 GRU 的结构 13
图 3.1 BERT 的输入 15
图 3.2 基于预训练的基础模型 16
图 3.3 基于 ERNIE 的 BiGRU 的模型结构 17
图 3.4 基于 ERNIE 的 TextCNN 的模型结构 18
图 3.5 基于 ERNIE 的 RCNN 的模型结构 19
图 4.1 对话标签数量分布 21
图 4.2 对话长度分布 21
图 4.3 数据集划分比例 23
图 4.4 模型训练流程图 24
图 4.5 卷积核个数和 BiGRU 隐藏单元实验结果 26
图 4.6 不同学习率模型实验结果 26
图 4.7 基于 BERT 及其变体的基础模型实验结果 27
图 4.8 基于 ERNIE 的改进模型实验结果 27
图 4.9 基于 ERNIE 的改进模型结果曲线 28

表目录

表 1.1 用户不规范表述 3
表 3.1 基于预训练模型的分类方法 15
表 4.1 领域标签标注准则 20
表 4.2 混淆矩阵 22
表 4.3 实验环境 23
表 4.4 预训练模型细节 24
表 4.5 基础模型训练参数 24
表 4.6 ERNIE-BiGRU 训练参数 25
表 4.7 ERNIE-TextCNN 训练参数 25
表 4.8 ERNIE-RCNN 训练参数 25
表 4.9 稀疏数据筛选参数设置 28
表 4.10 数据增强参数设置 28
表 4.11 数据增强模型对比 29
表 4.12 各标签具体表现 29

基于真实对话日志的对话领域多分类的设计与实现

软件工程专业学生　　白振翰

指导教师　　刘红娟

摘要：随着自然语言处理技术的持续进步，对话系统已逐渐成为人工智能应用中的一大重要领域。对话领域多分类任务在对话系统的自然语言理解模块中占据重要地位，对系统的效能产生关键影响。针对中文对话领域对话系统面临的许多挑战，利用真实对话日志自主标注了中文对话多分类数据集，并且提出了一种基于预训练模型的端到端多分类解决方案。在此基础上，对表现优异的 ERNIE 基础模型进行了改进，并通过一系列对比实验，验证了模型的高效性和科学合理性。最后，通过数据增强技术，使得 ERNIE-RCNN 模型在测试集上的 Micro F1 得分达到了 88.99%，从而有效地解决了对话领域的多分类问题。

关键词：对话系统　真实对话　领域多分类　深度学习　ERNIE 模型

Design and Implementation of Multi-Classification in Dialogue Domain Based on Real Dialogue Logs

Student majoring in Software Engineering　　Bai Zhenhan

Tutor　　Liu Hongjuan

Abstract: With the continuous progress of natural language processing technology, dialogue system has gradually become an important field in the application of artificial intelligence. The multi-category task in the dialogue field occupies an important position in the natural language understanding module of the dialogue system, and has a key impact on the performance of the system. Aiming at many challenges faced by the dialogue system in the field of Chinese dialogue, the real dialogue log is used to independently label the Chinese dialogue multi-classification dataset, and an end-to-end multi-classification solution based on the pre-trained model is proposed. On this basis, the excellent ERNIE basic model was improved, and a series of comparative experiments verified the efficiency and scientific rationality of the model. Finally, through data enhancement technology, the Micro F1 score of the ERNIE-RCNN model on the test set reached 88.99%, thus effectively solving the multi-classification problem in the dialogue field.

Key words: Dialogue System；Real Dialogue Domain；Multiple Classification；Deep Learning；ERNIE Model

1 绪论

1.1 选题背景

近年来，随着自然语言处理（NLP）技术的不断发展，对话系统已逐渐成为人工智能应用的重要领域之一。目前市场上主流的对话系统包括 Siri Google、Assistant、Amazon Alexa 等。从设计目的上来看，对话系统可分为任务导向型和非任务导向型两种类型。

任务导向型对话（TOD）系统旨在帮助用户实现他们的目标，同时具有更高的商业价值，能够极大地帮助人们提升效率。对于基于真实对话日志的对话系统来说，它可以帮助公司迅速定位客户的关注领域、用户意图、用户情感等。因此，本文的研究聚焦于任务导向型对话系统。

任务导向型对话系统流程如图 1.1 所示。用户的语音输入首先经过自动语音识别（ASR）模块将其转换为相应的文本形式。接下来，这些文本信息会被送入自然语言理解（NLU）模块，以进行深入的语义分析。在 NLU 阶段，系统会对输入文本进行全面解读，以获取关键信息，如领域标签、意图标签和槽位信息等，从而为后续处理和响应提供有力支持。然后，将文本理解的结果传递给对话管理（DM）模块，该模块负责处理上下文理解、对话状态更新、决策生成和帧生成等操作。在这个过程中，系统还可以与外部 API 和数据库进行知识交互，以提供更丰富的信息支持。最后，通过自然语言生成（NLG）和语音合成（TTS）模块，将处理好的信息转换为最终的语音回复，实现与用户的智能交互。

但是，在实际使用中，由于用户表达的不规范性、多领域的混合性以及意图的潜在性等因素，语义理解的难度增加了。这无疑对对话系统的智能化程度提出了更高的要求，也使得语义理解模块发挥了越来越重要的作用。

自 21 世纪初以来，自然语言理解任务通常被视为一组子任务[1]：领域分类、意图识别、槽位填充。领域分类可以被视为分类问题，可以使用任何分类方法解决，其目的是将对话划分为指定的领域标签，标签数量为一个或多个。意图识别也可以被视为分类问题，将意图划分到指定的意图标签中。而槽位填充则是序列标注问题，可以使用序列标记方法。本文的重点研究是对话领域多分类问题。

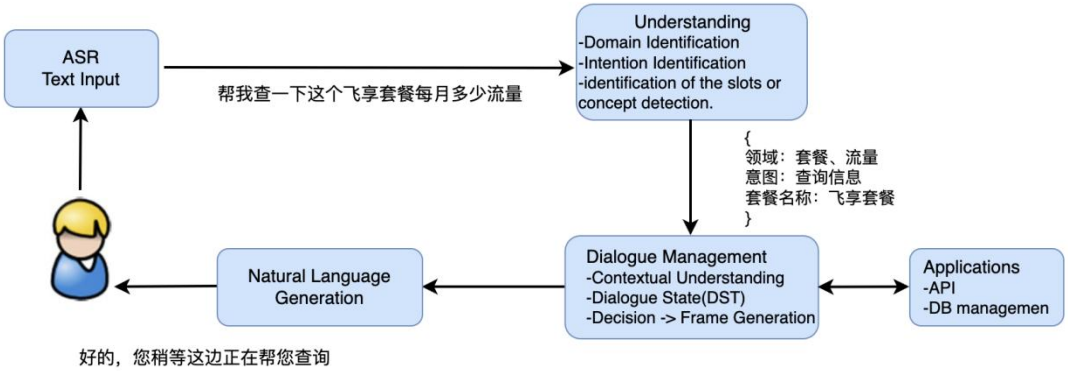


图 1.1 任务导向型对话系统流程

1.2 对话领域多分类的挑战

1.2.1 数据集的匮乏

近年来随着自然语言处理技术的发展，国内外许多大型互联网公司也都推出了对话助手，例如：苹果的 Siri、百度的 Duer 和阿里巴巴的天猫精灵等。但由于对话数据的隐私性以及对话系统的智能性不够，用户的信任度不高等原因。学术界可以使的大规模、高质量的中文公开数据集非常匮乏。这使得对话领域多分类具有一定的挑战性。

1.2.2 ASR 识别的不准确性

ASR 模块的主要作用是将用户的语音输入转换为文本，并将其输入到语义理解模块中进行后续处理。尽管近年来深度学习技术的发展已经显著提高了 ASR 的准确性[2]，深度神经网络（DNN）更是取代了传统的高斯混合模型，实现了从混合建模过渡到端到端的实现[3]。然而，即使在 DNN 的背景下，ASR 的准确性仍然难以达到预期，特别是在输入声音质量不佳的情况下，例如噪声干扰、口音问题、词汇语法问题等。在某些情况下，ASR 识别的不准确性也增加了对话领域多分类问题的挑战性。

1.2.3 用户表述的不规范性

在真实对话中，用户的表述往往容易口语化、方言化，这增加了模型理解语义的难度。例如，表 1.1 中序号 1 的表述：“我用积分换了那个 5G 的金币”，在真实对话中指的是“通过积分兑换获得流量奖励”。用户表述的不规范性对数据和模型都提出了更高的要求，这增加了对话领域多分类问题的挑战性。因此，对话系统需要更加灵活和鲁棒的模型来处理用户的口语化、方言化表述，以提高对话系统的准确性和可用性。

表 1.1 用户不规范表述

序号	对话表达方式	对话真实领域
1	我用积分换了那个 5G 的金币	积分换取流量的活动
2	三十块钱 5 个 G	三十块钱 5G 的流量套餐
3	宽带盒子、那个盒子	指机顶盒
4	帮我查一下本月花了多少钱	查询本月的账单
5	给我把这些乱七八糟的去掉	取消无用业务

1.2.4 对话多领域的混合性

在真实对话中，一些用户问题的复杂性可能导致对话时间特别长，进而使得对话文本的长度较长。本文采用的是对话级别的领域分类。然而，这面临着处理长文本和多标签分类问题的挑战。如何有效处理这些问题，以精确识别出对话所涉及的领域，是对话领域多分类的重要挑战之一。因此，对话系统需要采用能够处理长文本和多标签分类的模型，并结合合适的特征选择和模型优化方法，以提高对话领域多分类的性能和准确性。

1.3 国内外研究现状

在过去，文本分类方法大多基于传统的机器学习方法，例如朴素贝叶斯、支持向量机（SVM）、隐马尔可夫模型（HMM）、随机森林和最近邻（KNN）等[4–6]。然而，要实现这些方法的良好性能，需要进行复杂且繁琐的特征工程。此外，这些方法的设计往往仅针对特定领域的特征，而在处理新任务时可能并不适用。

随着深度学习的不断发展，过去的局限逐渐被克服。现代文本分类方法采用神经网络技术来自动从文本中学习并提取有效特征。在 2013 年，Mikolov 等人提出了一种名为 Word2Vec[7]的模型，它能够将词汇转化为计算机可处理的形式。因此，深度学习模型如卷积神经网络（CNN）、循环神经网络（RNN）及其变体在处理自然语言处理（NLP）任务时表现出了极大的灵活性和高效性能[8–12]。这些神经网络技术的成功应用为自然语言处理领域带来了革命性的变革，使得处理各种任务时能够更好地适应不同领域，从而实现了显著的性能提升。

2017 年 Google 提出了 Transformer[13]架构，对 NLP 领域产生了巨大影响，极大的推动了 NLP 技术的发展和进步。自 2018 年以来，大规模的基于 Transformer 的预训练语言模型(PLM)逐渐兴起，如 BERT[14]，OpenGPT[15]。2023 年 3 月 OpenAI 发布的 GPT-4[16]更是被证明能力接近人类，可以被认为是一个通用人工智能的版本[17]，人工智能的时代已经来临。

相比于基于 CNN 或 LSTM 的上下文嵌入模型，基于 Transformer 的预训练语言模型采用了更深层次的网络架构，并在大规模无监督文本上进行预训练。这种预训练模型的优势在于，它们可以为各种 NLP 任务提供强大的上下文嵌入表示，而不需要大量的特定任务训练数据。同时 Transformer 具有更强的语义特征抽取能力，在能够捕获更长距离的特征同时拥有更高的并行计算能力以及运算效率[18,15]。

预训练（Pre-training）最初被应用于计算机视觉（CV）领域。它可以使用大规模未标注的数据来训练模型，并将预训练的模型应用于特定任务中进行微调，以提高性能。预训练的主要优点包括加速模型训练和提高模型性能，尤其在数据集较小的情况下特别有用。

2009 年，邓嘉和李飞飞在 CVPR 上发表了 ImageNet[19]的论文。ImageNet 迅速成为有影响力的竞赛，目标是通过分类物体，选出错误率最低的算法。它解决了深度学习领域的数据不足问题。计算机视觉中，ImageNet 预训练使得网络学习通用视觉特征，提升其他任务性能。这凸显了大规模数据集在深度学习中的作用。

图像领域的预训练成功启发了 NLP 领域的应用，图 1.2 展示了 NLP 领域预训练模型的发展历程。预训练模型通过大规模的无监督学习，利用大量未标注的数据解决了数据稀疏问题。与此同时，模型学习的通用语言知识可迁移到下游 NLP 任务，从而提高下游任务的效果，并减少了人工特征工程的需求。预训练模型极大的推动了 NLP 领域的发展，为各项任务提供支持。本文关注预训练模型在领域多分类中的应用。

早期，使用预训练模型解决 NLP 领域问题的流程通常包括以下步骤：第一步，利用 Word2Vec 等预训练模型训练词向量。这些词向量可用作神经网络第一层参数的初始化。第二步，在训练过程中，利用 CNN、RNN 等神经网络结构对

下游任务参数进行更新。

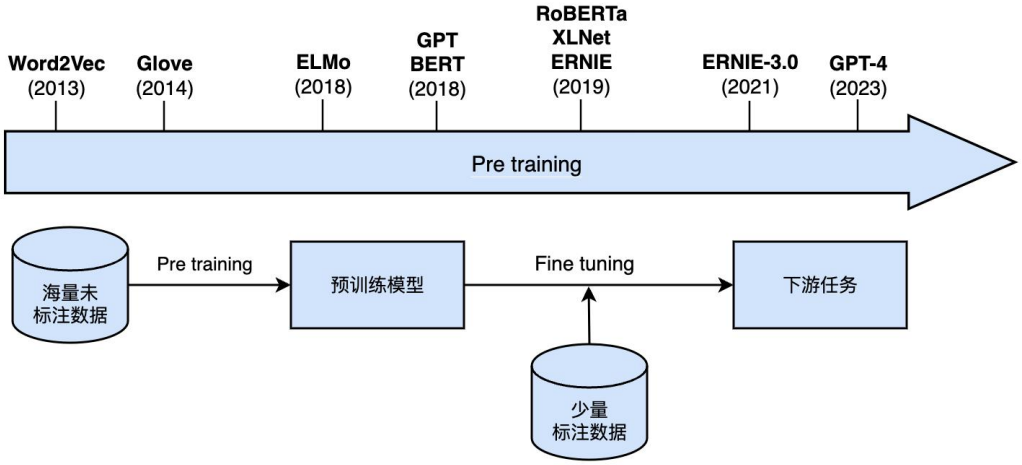


图 1.2 预训练模型历史

然而，Word2Vec 的主要问题是无法区分多义词的不同含义。为解决此问题，ELMo[20]提出了一种简洁有效的方法，通过考虑上下文动态调整词向量。ELMo 使用双向语言模型（包含两个 LSTM）分别从左至右和从右至左对单词进行编码。

尽管 ELMo 的双向预训练有一定优势，但它仍不完全，并且需要针对下游任务设计特定的网络结构。BERT 的出现解决了这种局限性，BERT 通过利用 Transformer 编码器在大量无标注文本语料库上进行预训练，并在特定任务上进行微调，从而生成高质量的文本表示。相较于过去的自然语言处理模型，BERT 在诸如文本分类、问答、阅读理解和序列标注等多个任务中表现出优异的性能[21–24]，大幅度提升了模型表现。之后预训练+微调的范式成为了 NLP 领域的主流方法。

然而，BERT 在学习词汇、短语和实体的完整语义方面存在一定局限。为此，百度推出了 ERNIE[25]。它对 BERT 的 MASK 策略进行了改进，采用了知识掩蔽策略，并加入了对话语言任务。在中文处理任务上，ERNIE 的表现超过了 BERT，展现出更优秀的性能。

1.4 主要工作及意义

本文的研究目标是使用预训练模型来解决任务导向型对话系统中的领域多分类问题。通过对比不同预训练模型，并对效果较好的基础模型进行改进，以获得更好的分类效果。

主要工作具体如下：

（1）本文概述了当前对话领域多分类任务所面临的挑战，并调查了国内外文本分类算法与 NLP 领域的研究现状和进展，以便能够更好地解决领域多分类任务。

（2）本文基于公开数据集，自主标注了 2000 条有关真实对话日志的中文对话，经数据增强最终共得到 5550 条对话。

（3）本文使用预训练模型进行了迁移学习实验，并且对基础模型进行改进。经多轮对比实验，验证了模型的有效性和科学性。最终模型在本文构建的中文对

话数据测试集的 Micro F1 score 为 88.99%，Macro F1 score 为 83.45%。

1.5 组织架构

本文在第一章介绍了任务导向型对话系统的流程以及选题背景，对目前领域内多分类问题所面临的挑战进行了深入探讨，接着梳理了文本分类算法和自然语言处理领域的研究现状。最后，明确了本研究的主要工作与意义，并对整篇文章的组织结构进行了总结概括。

第二章详细介绍了解决领域多分类问题所涉及的相关技术和理论。首先，讨论了词嵌入模型 Word2Vec，以及 Seq2Seq 和注意力机制。随后，深入探讨了 Transformer 架构以及预训练语言模型 BERT 及其各种变体。最后，简要概述了 CNN、RNN 及其相关变体的基本理论知识。

第三章的内容主要集中在使用预训练模型解决领域多分类问题。首先，本文对预训练模型处理领域多分类问题的流程进行了详细介绍，并提出了如 BERT-base 等基础模型。接下来，提出了三种改进模型：ERNIE-BiGRU、ERNIE-TextCNN 和 ERNIE-RCNN，通过对比分析来选定表现最佳的模型。最后，针对训练数据不足的问题，本文提出了一种数据增强的解决方案。

第四章基于第三章提出的模型进行了实验设计。首先，本文构建了一个中文领域的多分类数据集。然后，本文进行了基于预训练模型的多分类实验，并详细对比分析了结果，以验证本文提出的模型的有效性和科学性。

第五章总结了本文的工作并对下一步的工作进行了展望。

2 相关技术与理论基础

Word2Vec 作为自然语言处理任务中常用的词嵌入模型，已经在众多应用中证明了其价值。然而，随着基于 Transformer 架构的预训练模型的出现和发展，如 BERT 和 ERNIE 等，许多 NLP 子任务在迁移学习过程中可以取得更优秀的效果。在处理下游任务时，通过采用 CNN、RNN 及其变体等神经网络结构与预训练模型（如 BERT、ERNIE 等）相结合的方法，可以进一步提高任务的性能表现，从而在各种自然语言处理任务中取得更好的成果。

2.1 Word2Vec 词向量的表示

Word Embedding 的历史可以追溯到 20 世纪 80 年代[26]。在早期，研究人员通常使用基于词频的表示方法，如 one-hot 编码。但这种方法存在一种明显的缺点就是无法捕捉到单词的语义关系。

随着神经方法的发展，研究人员开始探索将单词表示为低维连续向量的方法，这就是现在所称的 Word Embedding，它将自然语言中的单词转化成向量的形式来方便计算机对自然语言进行处理。

Word2Vec 是实现 Word Embedding 的方法之一，通过自监督 Embedding 学习来生成与任务无关的词向量。如图 2.1 所示，Word2Vec 包括两种实现方式：CBOW 和 Skip-gram。

CBOW 模型通过上下文词预测中心词。输入为上下文单词向量，输出为中心词向量。CBOW 使用多层神经网络，旨在最小化中心词预测误差，从而学习词之间的关系。其训练速度较快，对低频词表现优异。

Skip-gram 模型则相反，通过中心词预测上下文词。输入为中心词向量，输出为上下文词向量。Skip-gram 也使用多层神经网络，目标是最小化上下文词预测误差。与 CBOW 相比，Skip-gram 更适合处理复杂语义词汇，但训练代价较高。

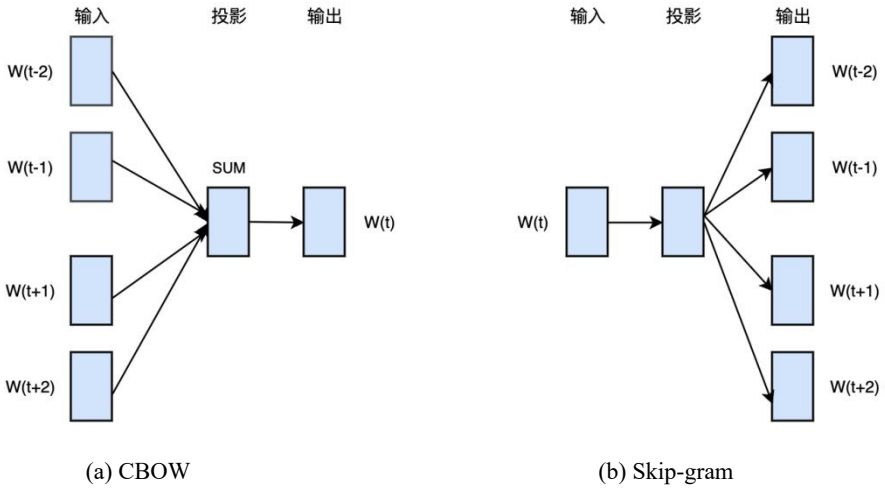


图 2.1 Word2Vec 的两种实现方式[7]

虽然 Word2Vec 是一种有效的单词嵌入模型，预训练的 Word Embedding 与传统方法相比具有更好的效果，但它存在着一些缺点，比如：

（1）无法处理多义词：因为对于同一个单词，Word2Vec 生成的 Embedding 都是相同的。
（2）无法捕捉所有的语义关系：Word2Vec 的上下文窗口固定，对于长文本序列，可能无法捕捉到所有的语义关系。

2.2 预训练模型

自从 Transformer 框架问世以来，基于 Transformer 架构的预训练模型已逐渐成为 NLP 领域的核心技术。在本部分中，将依次介绍 BERT、ERNIE 等预训练模型所采用的相关技术。

2.2.1 Seq2Seq

Seq2Seq 模型[27]由 Sutskever 等人于 2014 年提出，最初用于机器翻译。如图 2.2 所示，该模型由两个循环神经网络组成：编码器和解码器。

Seq2Seq 模型将一个序列映射到另一个序列，适应输入和输出序列长度不同的情况。编码器将输入序列编码成固定长度向量，解码器以此为输入生成输出序列。

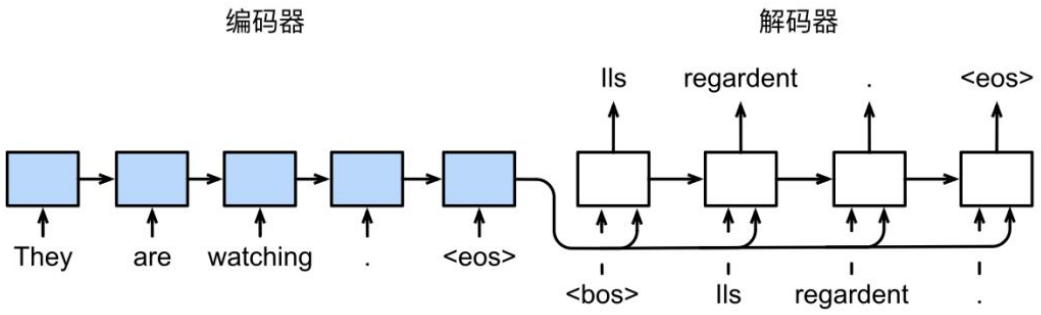


图 2.2 Seq2Seq 架构

Seq2Seq 模型的训练方式可以分为编码器和解码器两个阶段。如公式(2.1)和(2.2)所示，在编码器训练过程之中将输入序列$x = (x_1, x_2, ..., x_T)$中的每个单词x_t输入到 RNN 中得到隐藏状态向量h_t。编码器的最后一个隐向量作为编码器输出的上下文向量c，输入到解码器中。

$$h_t = EncoderRNN(x_t, h_{t-1}), t \in [1, N]\#(2.1)$$

$$c = h_T\#(2.2)$$

在解码器阶段，使用另一个 RNN 对目标序列进行生成。将公式(2.2)得到的上下文向量c作为初始隐藏状态向量s_0，并使用当前时间步的输入y_t和前一个时间步的隐藏状态s_{t-1}，生成当前时间步的隐藏状态s_t。然后，将s_t和上下文向量c作为输入，生成当前时间步的预测输出$\hat{y}_t$，即：

$$s_t = DecoderRNN(y_t, s_{t-1}), t \in [1, N]\#(2.3)$$

$$\hat{y}_t = softmax\big(f(s_t, c)\big), t \in [1, N]\#(2.4)$$

其中，f是一个非线性变换，将s_t和c转换成预测输出的概率分布。

2.2.2 注意力机制

2014 年 Bahdanau 等人提出注意力机制[28]。注意力机制是深度学习中常用的一种技术，其主要思想是在处理大量信息时，将注意力集中在最重要的信息上，即筛选出少量的关键信息并对其进行深入分析。这类似于人类大脑在处理信息时的工作方式，能够有效地提高深度学习模型的性能和效率。

注意力机制的提出动机是：在使用传统的 Seq2Seq 架构进行机器翻译时，Encoder 输入序列的每个单词对 Decoder 的下一个状态的重要性是不同的。因此模型计算了每个输入位置的注意力权重来表示当前位置的重要性，使得 Decoder 更加关注当前状态相关的信息，在计算上相当于对 Encoder 的输出重新进行了一次加权计算。Attention 的模型结构如图 2.3 所示。

注意力机制在机器翻译中获得了更好的效果，并且解决了随着输入句子长度的增加，Seq2Seq 的性能迅速恶化的问题。此后，注意力机制被广泛应用于各种 NLP 任务，例如机器翻译、问答系统和文本摘要等。

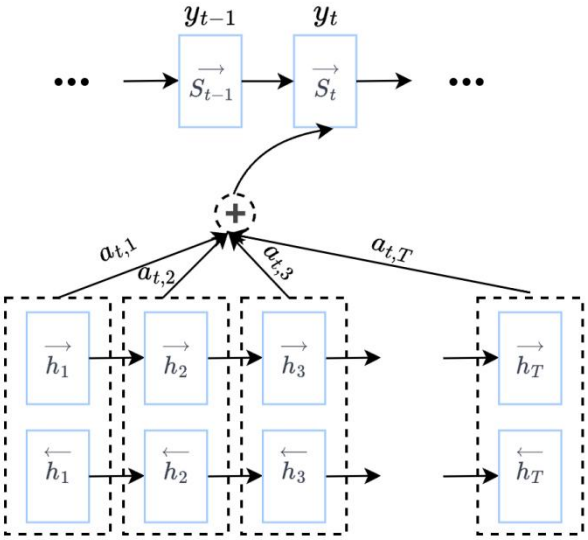


图 2.3 Attention 的模型结构[28]

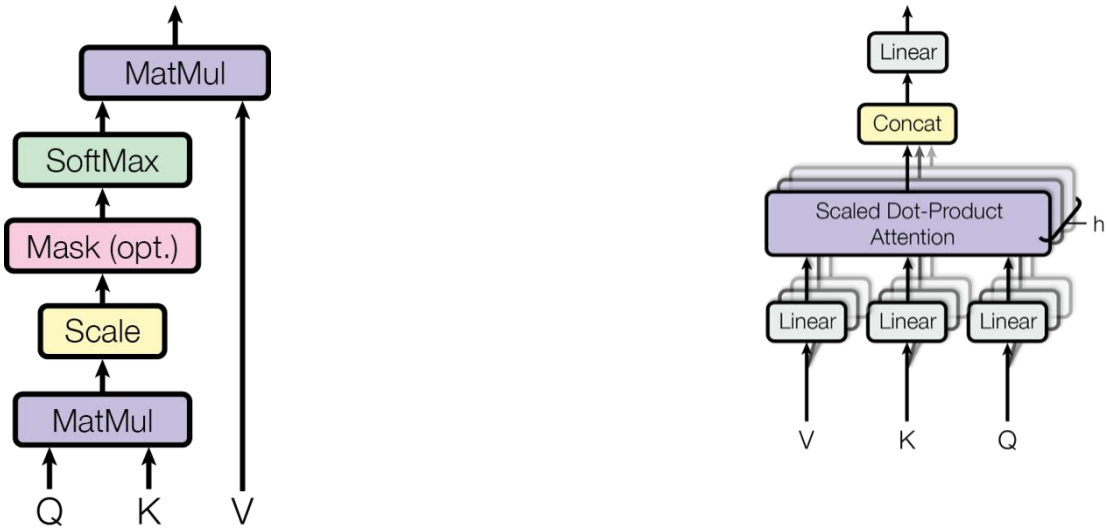
如公式（2.5）所示，注意力机制的计算过程中使用$query$来对$< key, value >$进行检索计算相似度得分，之后将计算到的相似度得分经过注意力评分函数α进行一定的处理，最后经过$softmax$运算进行归一化得到注意力权重，这些权重就是$value$在当前任务中的重要性，接着将$value$按照权重进行加权计算得到注意力机制的输出。

$$f\big(q,(k_1,v_1),\ldots,(k_m,v_m)\big)=\sum_{i=1}^{m} softmax(\alpha(q,k_i))v_i \#(2.5)$$

自注意力机制是$query = \text{key} = \text{value}$，即同一组输入序列同时充当查询、键和值，结果相当于对输入序列进行了加权处理，输入与输出维度大小相同。Transformer 中的自注意力 attention 如公式(2.6)所示，其中，Q、K、V 分别表示查询、键、值的矩阵形式，d_k表示每个查询和键的向量维度。$softmax$函数对 Q 和 K 点积得到的分数矩阵归一化后，与 V 相乘以得到最终自注意力向量。

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \#(2.6)$$

如图 2.4 所示，单头自注意力采用缩放点积注意力评分函数，而多头自注意力机制则由多个单头注意力层组成。多头自注意力能够学习更多信息，因为每个头可以关注不同的特征和语义信息。在最后阶段，各个头学到的信息被拼接成一个整体表示，这有助于捕捉序列中更丰富的信息和关系。



（a）单头自注意力　　（b）多头自注意力

图 2.4 单头和多头自注意力机制[13]

2.2.3 Transformer 框架

与传统的 CNN、RNN 等结构相比 Transformer 架构是完全依赖于自注意力机制的，并且在传统的 Encoder-Decoder 架构基础之上进行了改进。如图 2.5 所示，右边是模型的 Decoder，左边是模型的 Encoder。

Transformer 架构较 RNN 架构而言能够进行高效并行运算，较 CNN 而言可以更好的解决长序列依赖关系，从而更适合自然语言处理任务。但由于没有使用 CNN 或 RNN，Transformer 引入了位置编码，为每个单词添加了一个位置向量，使得模型能够处理任意长度的序列来更好的捕获单词中的依赖关系。

Transformer 架构使用了残差连接（Residual Connection），该思想来自 2015 年的 ImageNet 挑战赛上何恺明等学者提出的 ResNet[29]。残差连接有助于训练更深层次的模型，因为它能够使得梯度在深度网络中更容易地传播。

此外，Transformer 中使用了层归一化（layer normalization），其可以加速训练过程并提高模型的泛化能力，因为它将输入数据缩放为平均值为 0、方差为 1 的分布，从而降低了不同层之间参数尺度差异的影响。

Transformer 架构中 Encoder 和 Decoder 部分均由六层相同的层堆叠而成。Encoder 的主要组成部分是多头注意力机制（multi-head self-attention）和前馈神经网络（feed-forward neural network）。每层通过多头注意力机制捕捉语义，再使用前馈神经网络对自注意力子层进行非线性变换。通过堆叠这些层，模型可以更好地捕捉序列中的信息，每层的输出会成为下一层的输入。

与 Encoder 不同的是 Decoder 的输入使用的是遮蔽多头注意力机制（Masked multi-head self-attention）。因为 Transformer 最初是用来做机器翻译的，在 Decoder 的过程之中不能看到后面的单词。

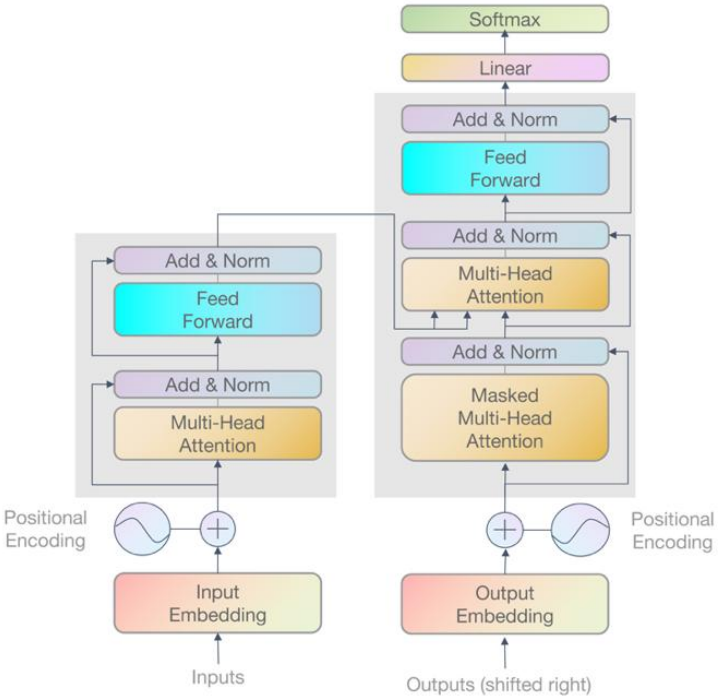


图 2.5 Transformer 框架[13]

2.2.4 BERT 及其变体

BERT 语言模型的成功开创了 NLP 领域的新范式（Pre training + Fine tuning）。BERT 的模型结构如图 2.6 所示。

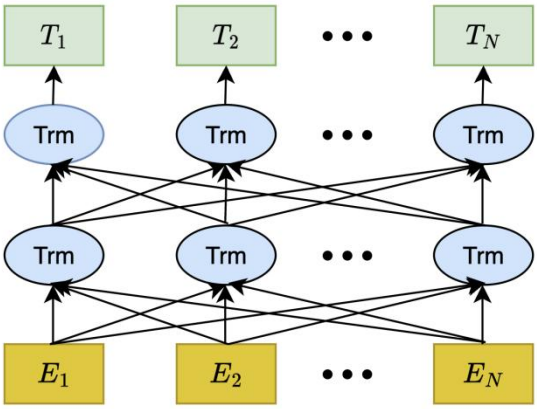


图 2.6 BERT 的模型结构[14]

ELMo 使用双向 LSTM 对输入序列进行编码，但由于每个单词的表示是根据整个序列计算的，对于长序列，ELMo 可能会受到记忆限制和计算效率等问题的限制。GPT 是一种基于 Transformer 的预训练语言模型，它使用单向的 Transformer-Decoder 对输入序列进行编码，并使用自回归的方式逐个预测下一个

单词。但是由于它是单向的，可能会影响其对上下文的理解能力。

BERT 是一个基于 Transformer 构建的多层双向 Encoder 网络，其最大特点在于利用双向语言模型进行表示学习。在自然语言处理任务中，BERT 的训练分为预训练阶段和模型微调阶段。

（1）预训练阶段：BERT 的预训练阶段包含两个子任务——遮蔽语言模型（MLM）和下一句预测（NSP）。在 MLM 任务中，BERT 通过随机遮蔽输入文本的部分单词，让模型根据上下文预测被遮蔽的单词，以学习语义信息。而 NSP 任务则让 BERT 学习句子间关系，作为一个二分类任务，有助于提高其在自然语言生成（NLG）任务上的性能。

（2）模型微调阶段：在处理具体的下游任务时，可以利用少量标注数据，在预训练好的 BERT 模型基础上进行微调。这种方法使得模型能够在较短时间内收敛并达到更好的性能，避免了从头开始训练的需求。通过这一阶段的微调，BERT 可以更好地适应并解决各种自然语言处理任务。

在改进 BERT 方面有许多工作。RoBERTa[30]比 BERT 更健壮，使用了更长的预训练时间和更多的训练数据，并且移除了 NSP 任务。XLNet[31]集成了 OpenGPT 自回归和 BERT 双向上下文建模的思想。XLNet 采用了 Transformer-XL[32]中的相对位置编码方法，使用了 Permutation Language Modeling（PLM）任务，在能够在处理更长的输入序列的同时对文本中的全局依赖关系进行更好的建模。

ERNIE 在中文语境下的表现得到了显著改进，处理中文任务效果更佳。ERNIE1.0[25]采用了字、短语和实体为单位的三种 mask 策略，并增加了 DLM（Dialogue Language Model）任务，使其在处理中文任务上的性能超过 BERT。ERNIE2.0[33]采用多任务学习方法，同时训练多个任务以提高泛化能力。该版本还提出了 word-aware、structure-aware 和 semantic-aware 三种无监督类型任务，在中英文任务中的表现全面超越 BERT。ERNIE3.0[34]采用了 Transformer-XL 结构，在百亿级预训练模型中引入大规模知识图谱，并使用 4TB 大规模语料进行联合掩码训练，从而大幅提升模型的知识记忆和推理能力。

2.3 基于深度学习的分类方法

2.3.1 卷积神经网络

基于卷积神经网络架构的模型已经在计算机视觉领域占据了主导地位，核心点是卷积具有局部性和平移不变性。随着词向量的提出，CNN 也可被应用于 NLP 任务中。

在文本分类任务中 CNN 可以快速捕获局部特征，同时保留特征的重要性，使模型在分类任务中表现出较好的性能。具体而言，在卷积层中，CNN 使用一维卷积对文本词向量进行卷积操作，生成包含局部信息的新向量。接下来，在池化层中，对卷积得到的特征进行池化操作来进一步压缩特征向量并保留重要信息。最后，通过全连接层将压缩后的特征向量映射到目标标签空间中，实现文本多分类任务。

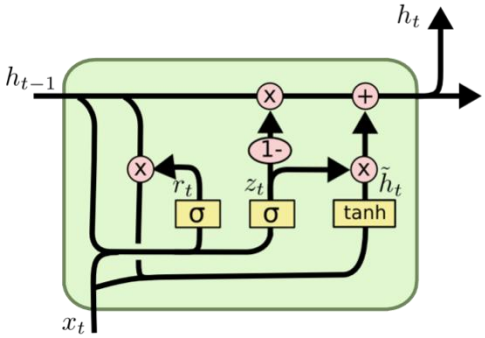
2014 年 Kim 提出了 TextCNN[12]，该模型的基本思想是使用大小不同的卷积核对文本不同的 n-gram 特征进行卷积操作，从而获得文本的特征表示。2017 年 Johnson 等提出 DPCNN[8]，其主要思想是利用深度金字塔结构和残差网络来提取文本特征，并通过池化操作将特征映射到固定长度的向量上去，最后接全连接层

进行分类。

2.3.2 循环神经网络及其变体

相较于 CNN，RNN 在处理序列任务上具有天然优势，这得益于 RNN 通过隐藏状态记忆先前信息，结合当前输入处理下一状态时生成新状态和输出。然而，在处理长序列时，RNN 存在梯度消失和梯度爆炸问题，导致模型难以学习长期依赖关系。

为解决以上问题，LSTM 采用门控机制，包括输入门、遗忘门和输出门，从而捕捉长序列依赖关系。如图 2.7 所示，GRU 作为 LSTM 变体，也使用门控机制控制信息流动。GRU 包括重置门和更新门，分别决定遗忘信息和更新信息程度。这使得 GRU 在处理长序列任务上优秀，且具有较少参数，降低计算复杂度。



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

图 2.7 GRU 的结构

2.4 本章小结

本章主要介绍了领域多分类任务中所用到的自然语言处理技术。首先介绍了 Word2Vec 词向量技术，随后详细介绍了 Seq2Seq 和 Attention 等 Transformer 架构所使用的技术，并分析了它们的优势和不足之处，最后介绍了 BERT、ERNIE 等预训练模型。并对卷积神经网络 CNN、循环神经网络 RNN 及其变体 GRU 和 LSTM 进行了介绍。

3 基于预训练模型的分类方法

随着 BERT、ERNIE 等预训练模型的提出，预训练加微调已经成为了 NLP 领域的新范式。这些预训练模型学习到了通用的文本特征，使得在特定任务中进行微调能够在较少的训练数据上达到优秀的效果。此外，相比传统的 CNN、RNN 等模型，这些预训练模型在抽取特征和进行语义学习方面具有更强的能力。因此，本章提出了一种基于预训练模型的领域多分类方法，并对效果较好的基础模型进行了改进。

3.1 问题描述

本章研究的对话领域多分类问题可视为文本多标签分类问题，与单标签分类不同多标签分类将每个样本划分到多个类别中。

假设存在 N 个样本，每个样本有 K 个可能的标签，对于第 i 个样本，其对应的标签集合为$y_i = y_{i1}, y_{i2}, ..., y_{iK}$，其中$y_{ik} \in 0,1$ 表示第 i 个样本是否具有第 K 个标签。多标签分类问题就可描述为：

给定N个样本的输入特征$X = x_1, x_2, ..., x_N$，通过学习一个函数$f(X)$将每个样本映射到其对应的标签集合$Y = y_1, y_2, ..., y_N$，$f(X)$可表示为$\mathrm{f}(X) = \{\mathrm{f}(X_1), f(X_2), ..., f(X_N)\}$。其中$f(x_i) = f_{i1}, f_{i2}, ..., f_{iK}$，$f_{ik}$表示第$i$个样本对于第$k$个标签的预测值，最后通常用 sigmoid 激活函数将输出转化到[0,1]的范围，设置阈值进行标签筛选。

如公式（3.1）所示，本文的损失函数是 BCEWithLogitsLoss，其中n为样本数量y_i为第i个样本的真实标签（取值为 0 或 1），y_i为第i个样本的预测值，σ(·)表示 sigmoid 函数。

$$BCEWithLogitsLoss(x, w) = \frac{1}{n}\sum_{i=1}^{n}\left[\left(-y_i \cdot \log\sigma(x_i) - (1 - y_i) \cdot \log\left(1 - \sigma(x_i)\right)\right)\right] \#(3.1)$$

3.2 算法描述

本文基于预训练模型 BERT 和 ERNIE 等，设计了两种模型：基础模型和改进模型。基础模型在预训练模型的基础上添加线性变换层和 sigmoid 层进行分类任务。而改进模型则在预训练模型后拼接了 TextCNN、BiGRU、RCNN 等神经网络结构，进一步提升模型性能。尽管模型结构不同，但模型训练的流程相同，具体方法在表 3.1 中有详细描述。

模型的输入是真实对话中用户表述转化成的文本数据S，S由子序列$x = (x_1, x_2, ..., x_P)$组成，其中P是文本长度。模型的输出为领域标签集合Y_i。

在进行训练前，需要对文本数据 S 执行预处理步骤。首先，利用分词器（tokenizer）将对话文本转换为对应的 embedding 向量。对于超过最大输入长度的文本，进行截断处理；对于不足最大输入长度的文本，则在其后补 0，以满足固定长度要求。接着，对对话标签执行 one-hot 编码处理，以便于模型理解和学习。最后，根据预设的每批样本数目（batch size），将处理完毕的数据组织成 data loader 形式，为后续模型训练提供便利。

表 3.1 基于预训练模型的分类方法

输入： $L = \{S_i, Y_i\}$，$i \in [1, n]$，$model$ 其中，S_i为对话训练样本，n为样本数量，Y_i为对应的领域标签集合，$model$为定义的模型（预训练基础模型或预训练改进模型）

输出： 效果最好的模型、模型的 F1score、模型 Loss

训练流程： 模型的参数进行初始化、迭代次数$epoch = N$，当前迭代次数 x，每个$epoch$ 的迭代次数 $step$

While x < N or Not Early Stop **do**

  for j in 1 ,… step **do**

    从训练集中随机抽取$batch\ size$个数据，进行前向传播

    进行反向传播对模型参数进行更新

  end for

  计算模型的 Loss，使用优化器对梯度和模型的参数进行更新

end while

利用训练好的模型在测试集上对样本进行评估，计算 F1 score

3.3 基于 BERT 及其变体的基础模型

本文使用了 bert-base-chinese、hfl/roberta-wwm-ext、chinese-xlnet-base、ernie-3.0-medium-zh 作为基础模型。

这些预训练模型的输入输出具有相似的结构。以 bert-base-chinese 为例，模型的 Transformer 层数为 12 层，每层内部的隐藏单元大小为 768，多头自注意力头数为 12。

如图 3.1 所示，BERT 的输入由三个嵌入向量的和组成：Token Embeddings、Segment Embeddings 和 Position Embeddings。

Token Embeddings 负责将输入序列中的每个 token 转换为相应的向量表示，Segment Embeddings 则用于识别输入中不同句子的边界，而 Position Embeddings 则为输入序列中的每个 token 编码其位置信息。在输入序列的起始处添加一个特殊的[CLS]标记以表示整个序列的语义信息，在句子的结尾添加一个特殊的[SEP]标记来分隔各个句子。最后，BERT 将这些嵌入向量相加，作为输入，以便在下游自然语言处理任务中进行训练或推理。

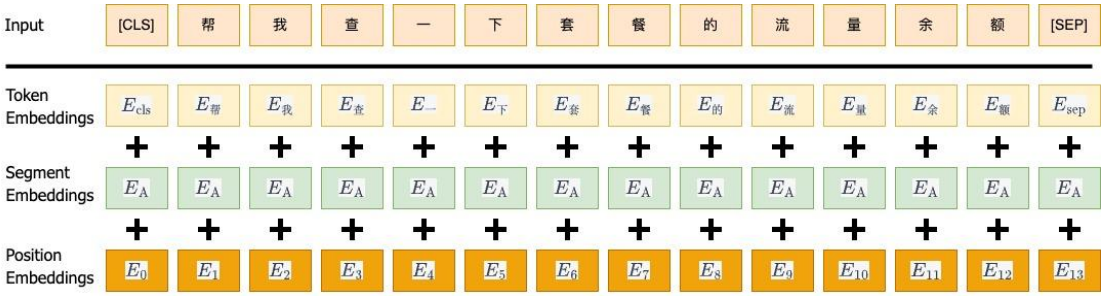


图 3.1 BERT 的输入

BERT 的输出可分为两种：pooler output 和 sequence output。

(1) pooler output 是一个形状为$[batch\ size, hidden\ size]$的向量，经过 BERT 的多层 Transformer 之后，使用最后一层中[CLS]标记对应的向量作为整个句子的表示，可以看作是 BERT 模型对整个输入序列学习到的所有信息的汇总。

(2) sequence output 是一个形状为$[batch\ size, max\ length, hidden\ size]$的向

量。在 BERT 中该向量是通过多层 Transformer 的 Encoder 来获取的。每层都会输出 sequence output，一般来说使用最后一层的 sequence output，因为该输出是对全局语义最好的学习汇总。在每层的 Transformer Encoder 中由于 Transformer 使用自注意力机制和前馈神经网络，输入输出的数据维度并不会改变。低层的 Transformer 学习到的更多的是局部的语义信息，这些底层特征往往是较为基础的特征，例如词的语义、语法等信息。高层的 Transformer 学习到的更多是全局的语义信息，例如句子的语义、主题等。

预训练基础模型解决领域多分类问题的模型结构如图 3.2 所示。将模型的 pooler output 输入到全连接层中，输出维度为$[batch\ size, num_labeles]$的向量，$num_labeles$为领域标签的个数。最后经过 Sigmoid 层获得每个标签的得分，根据阈值选出对应的标签，得分大于阈值则为认为样本属于该标签，反之则认为不属于。

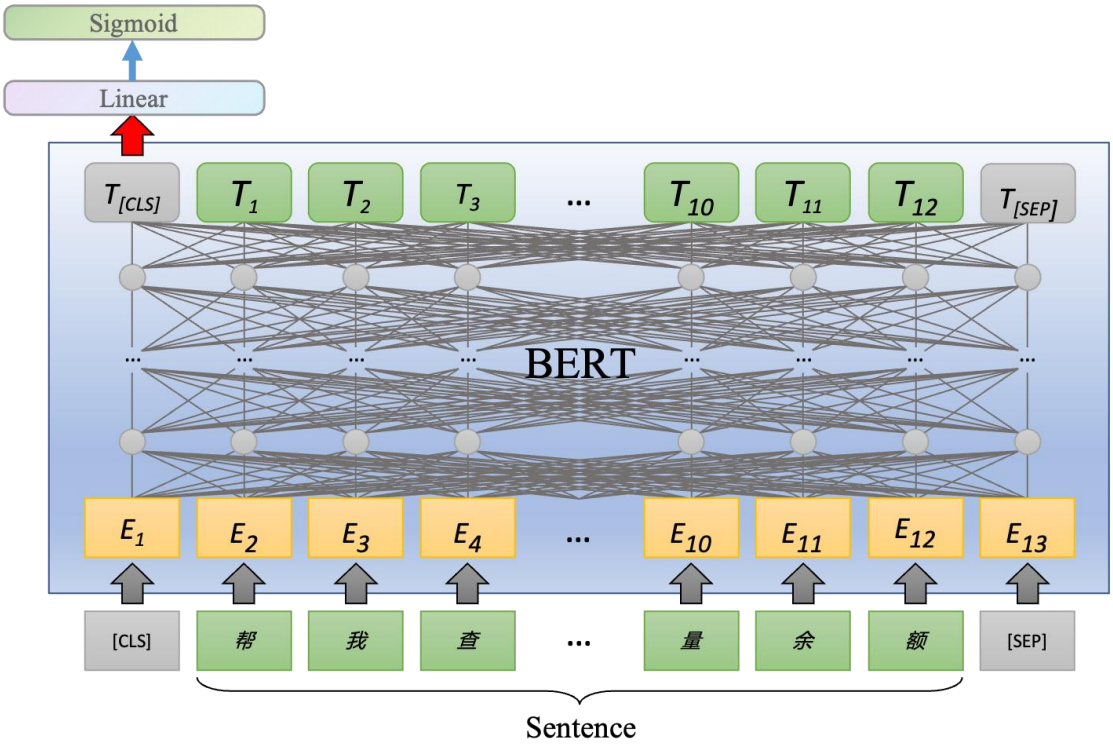


图 3.2 基于预训练的基础模型

3.4 基于 ERNIE 的改进模型

ERNIE 的输出与 BERT 基本相同。将 ERNIE 输出的 sequence output 与 CNN、RNN 等神经网络相结合的优势在于：这样可以充分发挥 ERNIE 的序列信息，而不是仅仅依赖于包含汇总信息的 pooler output 作为整个序列的代表。这是因为 pooler output 只进行了信息压缩，可能会导致一些有用信息的丢失。

通过将 ERNIE 的输出作为 CNN、RNN 等神经网络的输入，能够进一步提取对话句子中的上下文信息，这将有助于模型更好地理解和挖掘真实对话日志的含义。如此一来，模型在实际应用中能够表现出更优秀的性能。

3.4.1 基于 ERNIE 的 BiGRU 模型

在基于 ERNIE 的 BiGRU 模型中，ERNIE 输出的 sequence output 是对输入序列的每个 token 都进行了编码，可以看作是句子的表示。

BiGRU 是一种循环神经网络，它具有一定的记忆能力，能够对长序列中的上下文信息进行建模，基于 ERNIE 的 BiGRU 的模型在处理长文本时可以更好地捕捉文本中的上下文信息，模型结构如图 3.3 所示。将 ERNIE 输出的 sequence output 作为 BiGRU 的输入可以让 BiGRU 进一步学习到这个表示中的更多特征和信息，得到更加准确的句子的表示，进一步提升模型的性能。

ERNIE 输出的 sequence output 向量维度为$[batch\ size, max\ length, hidden\ size]$。将 sequence output 输入到 BiGRU 后，可以得到维度为$[batch\ size, max\ length, BiGRU\ hidden\ size\ *\ 2]$的隐藏状态。这是因为 BiGRU 是双向的，将正向和反向两个方向学习到的信息拼接，以便更好地学习语义。基于 ERNIE 的 BiGRU 模型从 BiGRU 输出中提取句子最后时刻的 hidden state，此时输出维度为$[batch\ size, BiGRU\ hidden\ size\ *\ 2]$。然后，通过线性变换层和 Sigmoid 层计算每个对话领域标签的得分，根据得分和阈值确定领域标签。

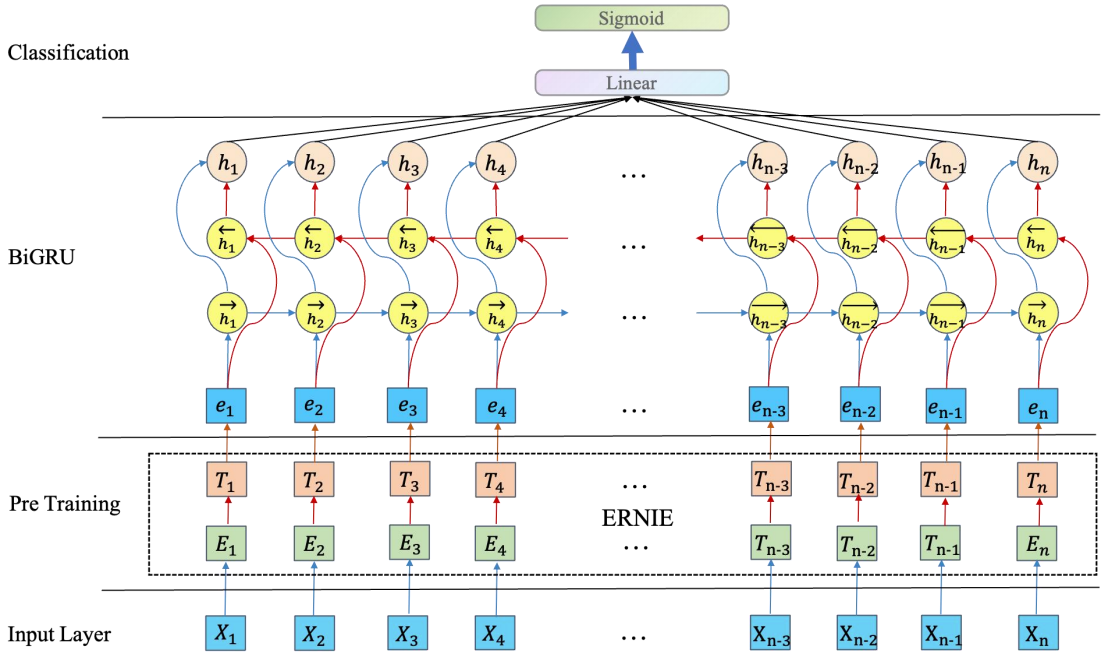


图 3.3 基于 ERNIE 的 BiGRU 的模型结构

3.4.2 基于 ERNIE 的 TextCNN 模型

基于 ERNIE 的 TextCNN 模型的优势在于其考虑到了输入文本的局部结构信息。对于真实对话日志来说，其包含了很多子串（如单词、词组、句子等）的组合，而不同的子串组合可能对应着不同的语义。TextCNN 能够在不同的子串上进行卷积和池化操作，从而捕捉到不同层次、不同长度的局部结构信息。这使得 TextCNN 能够更好地处理真实对话日志的领域多分类任务，同时避免了使用全局信息导致的过拟合问题。

在该模型中，同样使用了 ERNIE 输出的 sequence output 作为 TextCNN 的输

入序列，假设使用的卷积核大小为（2，3，4），卷积核数量为 512，输入序列的最大长度为 1024，模型结构如图 3.4 所示。

首先对 sequence output 增加一个维度来便于卷积操作，增加维度后的向量形状为$[batch\ size, 1, max\ length, hidden\ size]$。接下将该向量输入到卷积列表的每个卷积层，再经过激活函数层进行非线性变换。经过一维卷积之后，输出的最后一个维度变成了 1，去掉最后一个维度之后，输出向量的形状大小为$[batch\ size,\ kernel\ num,\ max\ len - kernel\ size\ +\ 1]$。然后对于每个卷积层的输出，都通过 MaxPool1D 池化层进行池化操作，并将所有池化结果拼接起来的到形状大小为$[batch\ size,\ kernel\ num\ *\ len(filter\ sizes)]$的特征向量。最终的特征表示向量被输出至线性变换层，接着经过 Sigmoid 层得到每个对话领域标签的得分，根据得分和阈值得到领域标签。

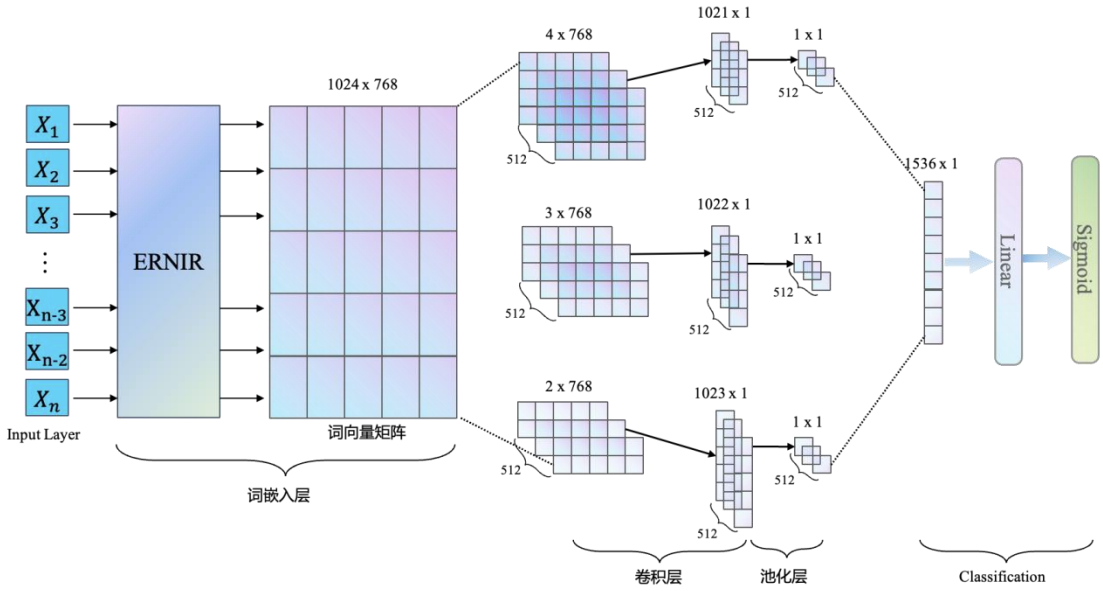


图 3.4 基于 ERNIE 的 TextCNN 的模型结构

3.4.3 基于 ERNIE 的 RCNN 模型

在基于 ERNIE-RCNN 的模型中，ERNIE 的 sequence output 作为 BiGRU 的输入序列，经过拼接后的向量维度为[batch size, max length, hidden size + BiGRU hidden size * 2]。随后，将最后两维进行交换，进行一维的最大池化操作，最后去掉一维，得到最终的特征表示。最终的特征表示将被输出至线性变换层，接着经过 Sigmoid 层得到每个对话领域标签的得分，根据得分和阈值就可得到领域标签。

与 ERNIE-BiGRU 不同的是 ERNIE-RCNN 将 BiGRU 的输出与 ERNIE 输出的 sequence output 拼接起来而不是直接使用 BiGRU 的输出作为后续的输入序列。

该模型的优势在于它不仅使用了 BiGRU 更好的捕捉了文本的上下文信息，而且将 ERNIE 学习到的信息与 BiGRU 学习到的信息进行了组合，最大程度上减少了信息的损失。Max pooling 的使用使得模型得到了最重要的信息。

假设 BiGRU 的隐藏单元数目为 300，输入序列的最大长度为 1024，模型结构如图 3.5 所示。

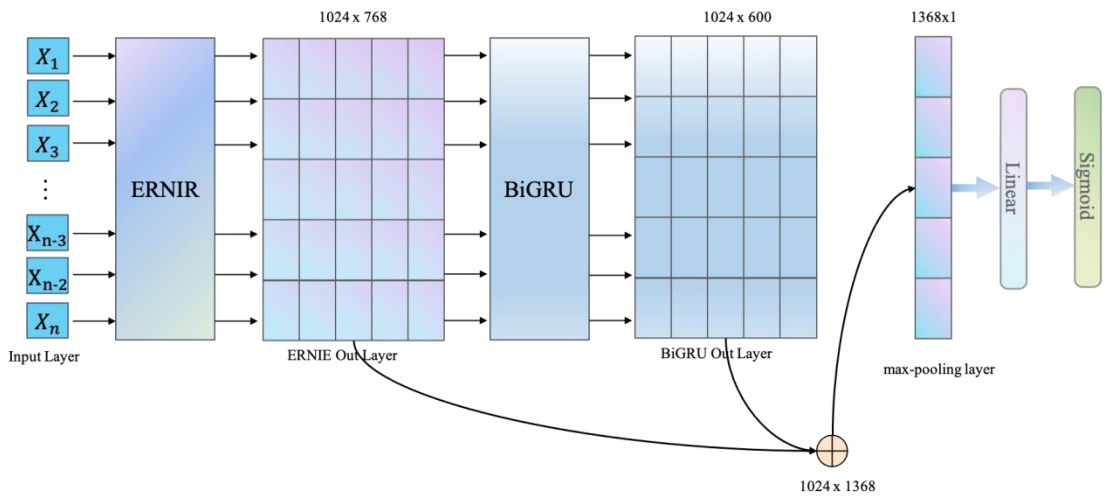


图 3.5 基于 ERNIE 的 RCNN 的模型结构

3.5 基于数据增强的改进模型

稀疏数据筛选适用于处理文本分类任务中的数据不平衡和训练数据覆盖不足问题。本文使用 TrustAI[1]来进行稀疏数据筛选。另外，为解决训练数据量较少的问题，本文采用 PaddleNLP 的内置函数[2]进行数据增强，通过诸如同义词替换、句子结构调整等方法，生成了更多含有多样性的训练数据。这些数据增强手段丰富了原始数据集，有助于提高模型的泛化能力和鲁棒性。借助这些额外生成的训练数据，ERNIE-RCNN 模型在处理不同类型的文本分类任务时可以取得更好的效果。

3.6 本章小结

本章对领域多分类问题进行了详细描述，并给出了使用预训练模型解决该问题的流程，介绍了基于 BERT 及其变体和基于 ERNIE 的改进模型的模型结构，并对模型的输入输出及优劣进行了详细说明。此外，本章还介绍了基于数据增强的改进模型。

[1] https://github.com/PaddlePaddle/TrustAI/actions
[2] https://github.com/PaddlePaddle/PaddleNLP

4 实验分析与讨论

在对话领域的多分类研究中，深度学习模型往往需要高质量的标注数据才能取得良好的效果。因此，本文基于公开竞赛数据集自主标注了高质量的对话领域分类数据集。

针对第 3 章中提出的模型，本章深入剖析了模型的参数细节和实验结果。在多组实验中，本文不仅仔细调整了模型参数，还采用了数据增强技术来提高模型的性能。通过对比实验结果，验证了本文所选模型的优越性和可行性。这些实验结果表明，所提出的模型具有较高的准确率和鲁棒性，有力地证明了本文所选模型在领域错分类任务中的有效性和实用性。

4.1 实验数据集

为了解决对话领域分类公开数据集不足的问题，本文使用了 MobileCS[3]（移动客户服务）公开对话数据集。然而，MobileCS 数据集中没有对话领域标签，因此本文自主标注了 2000 条中文对话数据集作为实验数据集。

这些数据的标注级别为对话级，即每个对话都有一个或多个领域标签。这些对话记录是真实用户与中国移动客服人员之间的真实对话记录，已进行隐私信息匿名处理。数据标注准则详见表 4.1。

在本章中，详细介绍了模型参数的细节和实验结果，并进行了数据增强实验以验证模型的优越性和可行性。基于现有主流业务需求，本文选取了在真实业务中出现较多的 11 个领域，包括“账单”、“流量”、“套餐”、“卡号”、“业务”、“宽带”、“机顶盒”、“工单”、“活动”、“用户信息”、“其他”。

实验数据集共包含 4731 个领域标签，平均每段对话有 2.4 个领域标签。对话的平均长度为 917 个字，最长对话长度为 6532 个字。有 1384 个对话长度小于 1024 个字。标签数量和对话长度的具体分布详见图 4.1 和图 4.2。

表 4.1 领域标签标注准则

领域标签	标注准则
账单	包含手机话费，账单，资费信息等
流量	流量查询订购，对流量使用存疑等
套餐	套餐的更换，套餐明细，费用咨询等
卡号	手机卡相关，手机信号问题等
业务	会员，彩铃等增值业务
宽带	宽带故障以及迁移等
机顶盒	电视机顶盒使用咨询或故障
工单	一些业务、故障问题或已经向客服登记过的问题进度咨询
活动	例如空中课堂，合约手机等线上线下推出的一些活动（非流量，套餐等优惠信息）
用户信息	涉及用户身份信息、家庭住址、手机号等
其他	一些未出现的问题统一标注为其他

[3] http://seretod.org/Challenge.html

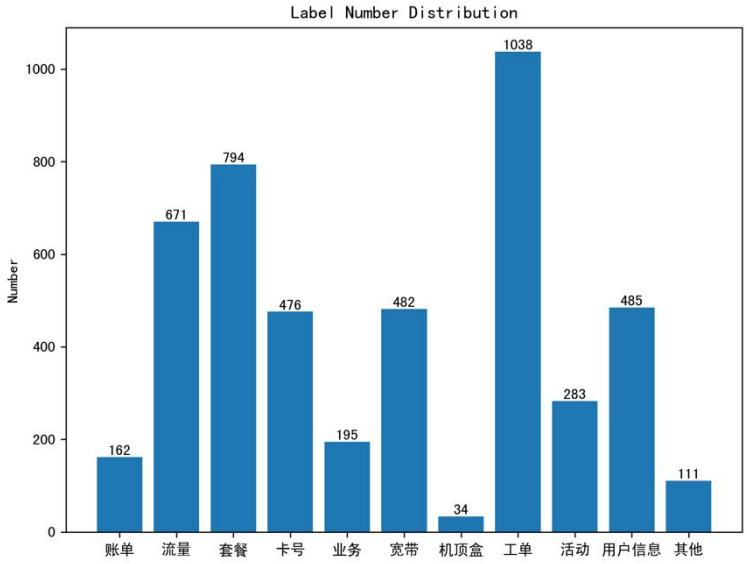


图 4.1 对话标签数量分布

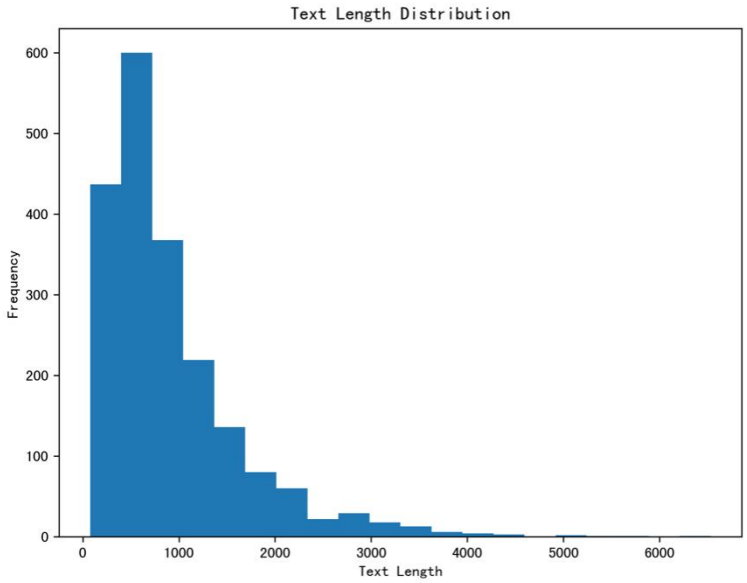


图 4.2 对话长度分布

4.2 评估指标

本文采用 Micro F1 score 和 Macro F1 score 来作为领域多分类任务模型性能的评估指标。

领域分类任务共有C=11 个类别，混淆矩阵如表 4.2 所示。TP_i表示预测为第i类且真实标签也为第i类的样本数量，FP_i表示预测为第i类但真实标签不为第i类的样本数量，FN_i表示预测不为第i类但真实标签为第i类的样本数量。

精确率（Precision）衡量的是模型在预测正例时的准确性，计算方法参见公式（4.3），而召回率（Recall）关注的是模型未能识别出的正例，计算方法参见公式（4.4）。精确率和召回率通常呈现出相互矛盾的关系，因为在分类时，模型需要在这两个指标之间进行权衡。F1 值则综合了精确率和召回率的影响，作

为一个综合评估指标。F1 值的范围在 0 到 1 之间，越接近 1 表示模型效果越好，F_1值的计算方法见公式（4.2）。

Micro F1 score 和 Macro F1 score 分别使用了微平均和宏平均方法来计算多分类任务的 F1 score。Micro F1 score 更多的关注的是全部标签的平均情况，每个标签的重要程度相同，而 Macro F1 score 更多的关注的是每个类别 F1 的平均值，每个类别具有相同的权重。Micro F1 score 和 Macro F1 score 的计算方法为公式（4.1）和（4.5）。

表 4.2 混淆矩阵

		预测类别	
实际类别		正例	反例
	正例	TP	FN
	反例	FP	TN

$$F_1^{macro} = \frac{1}{C}\sum_{i=1}^{C} F_1^i \#(4.1)$$

其中：

$$F_1^i = 2 \cdot \frac{P_i \cdot R_i}{P_i + R_i} \#(4.2)$$

$$\mathrm{P_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FP_i}} \#(4.3)$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \#(4.4)$$

Micro F1 score：

$$F_1^{micro} = \frac{2\sum_{i=1}^{C} TP_i}{2\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i + \sum_{i=1}^{C} FN_i} \#(4.5)$$

4.3 实验准备

4.3.1 实验环境

本文实验在 Linux 系统的服务器上进行，使用了 PaddlePaddle 深度学习框架。具体实验环境参数详见表 4.3。

PaddlePaddle 是百度开源的深度学习框架，采用高效的动态图计算方式，能够实现高速的训练和推理，同时也易于使用、高效、灵活，全面支持深度学习任务，例如语音识别、自然语言处理、图像分类、目标检测等。PaddlePaddle 的设计理念强调了易用性和扩展性，它提供了丰富的 API 和工具，使得开发者能够轻松地构建和训练复杂的神经网络模型。此外 PaddlePaddle 能够充分利用 GPU、TPU 等硬件设备进行加速计算，同时支持分布式训练，以便在大规模数据和模型场景下实现高效的模型训练。

表 4.3 实验环境

实验环境	参数
操作系统	CentOS Linux release 7.7.1908 (Core)
GPU	Tesla V100-SXM2-32GB
CPU	Intel(R) Xeon(R) Gold 6271C CPU @ 2.60GHz
CUDA	11.2
CUDNN	8.1.0
深度学习框架	PaddlePaddle 2.4.0
NLP 框架	PaddleNLP 2.4.2

4.3.2 预处理

(1) 文本预处理

为了后续的领域多分类任务更加准确，本文先通过自动化的方式对 ASR 转录后的 MobileCS 数据进行初步筛选，只选出语义流畅的 2000 条句子进行标注和训练。

本文使用百度提供的语言模型 zh_giga.no_cna_cmn.prune01244.klm[4]作为基模型，使用改进的 pycorrector 工具进行检错。检错依据包括字音困惑集、词频、困惑度、RNN 模型四个维度。

模型综合考虑了以上四个维度，对于一句话的错词数量进行评估。如果错词出现在字音困惑集中，或者词频较低、困惑度较高，都会提高 RNN 模型的阈值。最后，通过 RNN 模型计算出错概率，并选出 2000 条文本进行下一步标注处理。

(2) 数据集划分

本文将数据集按照 7:1:2 的比例划分为训练集、验证集和测试集。具体数据量分布为：训练集 1400 条，验证集 200 条，测试集 400 条。

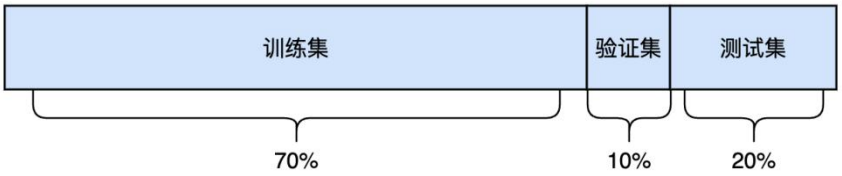


图 4.3 数据集划分比例

4.4 实验设计

在本部分中，首先进行了针对 BERT 及其变体的基础模型的对比实验。实验结果显示，ERNIE-base 在这些模型中表现最为优秀。接下来，以 ERNIE-base 为基础进行了改进，同时采用了数据增强技术，以进一步提升模型的性能。

如图 4.4 所示，在训练阶段步骤如下：

（1）读取数据并加载预训练模型，以进行模型参数的初始化。

（2）进行模型训练，网络参数不断进行调整。

[4] https://github.com/shibing624/pycorrector

（3）使用验证集实时评估模型性能。

（4）经过多轮迭代，以动态图方式保存性能最佳的模型参数，作为领域分类任务的最终模型参数。

在测试集评估阶段：

（1）重新实例化一个模型。

（2）将训练过程中保存的最终模型参数加载到新模型中。

（3）使用新模型对测试集进行评估。

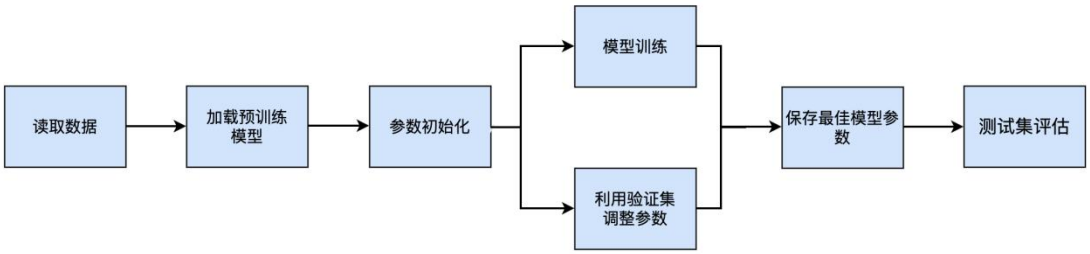


图 4.4 模型训练流程图

4.4.1 实验参数设置

(1) 基于 BERT 及其变体的基础模型

本部分选用 bert-base-chinese、hfl/roberta-wwm-ext、chinese-xlnet-base、ernie-3.0-medium-zh 做为的基础模型。百度于 2022 年 5 月开源 ernie-3.0-medium-zh。基础模型的细节如表 4.4 所示。

BERT 模型和 RoBERTa 模型最多只能接受文本长度为 512 的输入，XLNet 和 ERNIE-3.0 可接受文本长度为 1024 的输入，将模型的输入长度设置成模型能够输入的最大长度。其余参数见表 4.5。

表 4.4 预训练模型细节

模型名称	Transformer 层数	隐藏单元个数	自注意力头数	参数量
bert-base-chinese	12	768	12	108M
hfl/roberta-wwm-ext	12	768	12	102M
chinese-xlnet-base	12	768	12	117M
ernie-3.0-medium-zh	6	768	12	75M

表 4.5 基础模型训练参数

参数名称	参数值
epoch	30
batch size	16
optimizer	Adam
learning rate	5e-5

(2) 基于 ERNIE 的改进模型

本部分使用验证集来对超参数进行多次调整，并比较了卷积核数目（32，64，128，256，512）和 BiGRU 隐藏单元个数（200、250、300、350、400、450、500）对模型性能的影响，结果如图 4.5 所示。最终选择了最优的超参数，用于改进模型的对比实验。

在实验中，max length 被固定为 1024，batch size 被固定为 16。这是因为将 batch size 设置为2^n有利于充分利用 GPU，并且在本实验设备上，max length 固定的情况下 batch size 可以选择的范围较小。当 batch size 为 16 时，GPU 的显存使用率可以维持在 80%左右。

最后，在效果最好的模型 ERNIE-RCNN 基础之上进行了学习率的选择，以适配本文使用的 batch size。通过图 4.6 可以看出，当学习率为 5e-5 时，模型收敛更快且更平稳，在验证集上的表现也更好。

ERNIE-BiGRU、ERNIE-TextCNN、ERNIE-RCNN 的具体参数分别如表 4.6、表 4.7、表 4.8 所示。需要注意的是，激活函数和 dropout rate 为 ERNIE 后接神经网络的参数。

表 4.6 ERNIE-BiGRU 训练参数

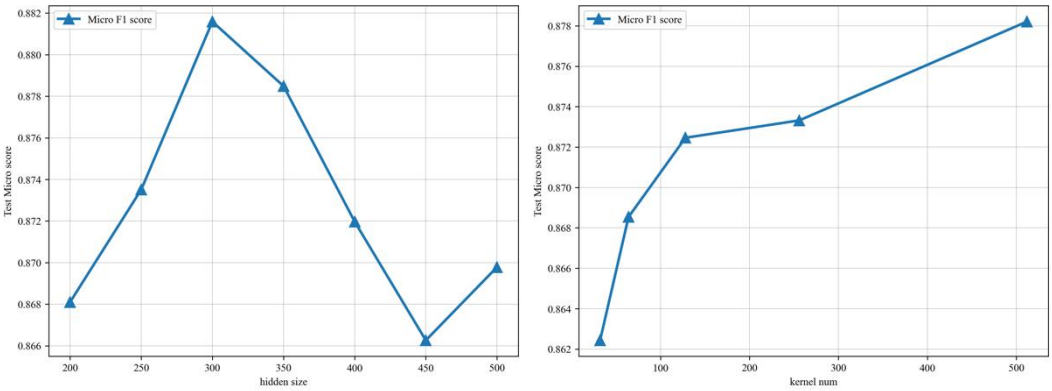
参数名称	参数值	参数名称	参数值
epoch	30	activation function	ReLU
batch size	16	BiGRU hidden size	300
optimizer	Adam	BiGRU layers	2
learning rate	5e-5	dropout rate	0.2

表 4.7 ERNIE-TextCNN 训练参数

参数名称	参数值	参数名称	参数值
epoch	30	activation function	ReLU
batch size	16	filter size	(2,3,4)
optimizer	Adam	kernel num	512
learning rate	5e-5	dropout rate	0.2

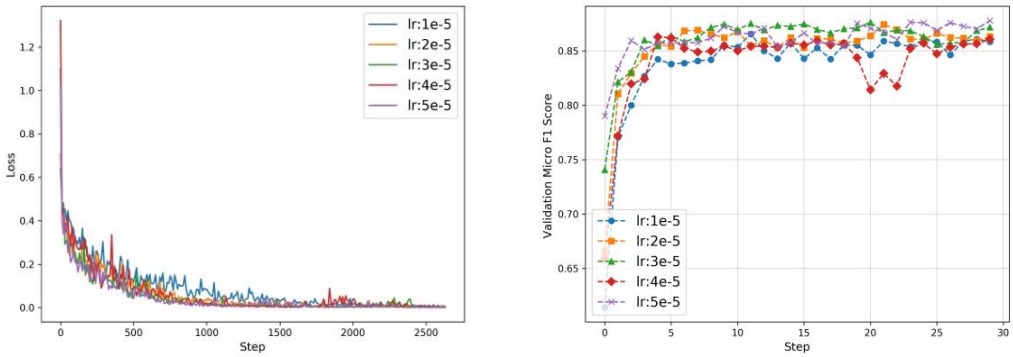
表 4.8 ERNIE-RCNN 训练参数

参数名称	参数值	参数名称	参数值
epoch	30	activation function	ReLU
batch size	16	BiGRU hidden size	300
optimizer	Adam	BiGRU layers	2
learning rate	5e-5	dropout rate	0.2
		polling	Max Polling



（a）BiGRU 隐藏单元实验结果　　（b）卷积核个数实验结果

图 4.5 卷积核个数和 BiGRU 隐藏单元实验结果



（a）不同学习率损失曲线　　（b）不同学习率验证曲线

图 4.6 不同学习率模型实验结果

4.5 实验结果对比分析

4.5.1 基于 BERT 及其变体的基础模型

在本部分的实验中，本文发现效果最优的基础模型是 ERNIE-base，其 Micro F1 达到了 87.01%。其次是 XLNet-base，之后是 BERT-base 和 RoBERTa-base，具体结果见图 4.7。

分析导致 BERT-base 和 RoBERTa-base 效果较差的主要原因可能是真实对话文本长度较长，最长的有 6532 个字。而这两个模型的最大输入只有 512 个字，超过 512 字的对话被截断，导致对话后面许多与领域标签相关的信息和语义没有被模型学习到。例如，在真实对话中，工单往往是一些需要客服登记的问题，而与工单有关的表述经常会出现在对话的最后，从而导致工单标签可能不被模型学习到。

虽然 XLNet-base 和 ERNIE-base 的最大输入均设置为 1024 个字符，但 ERNIE-base 的效果更好，说明 ERNIE 在预训练时引入的大规模知识图谱、使用的知识掩码策略以及更大的中文数据集等能够使得模型更加有效地对中文对话文本进行处理。

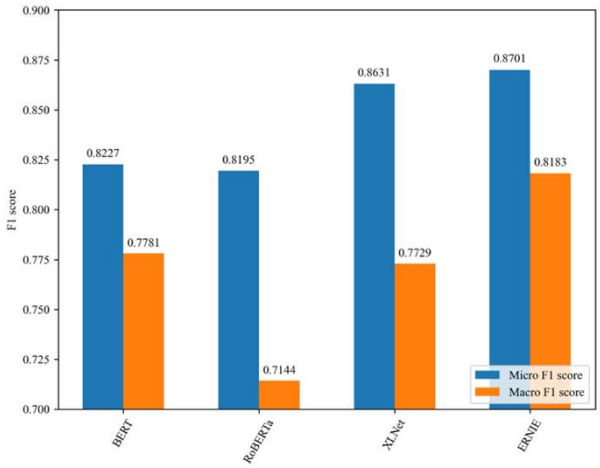


图 4.7 基于 BERT 及其变体的基础模型实验结果

4.5.2 基于 ERNIE 的改进模型

本部分选取了上一步中效果较好的 ERNIE 模型作为预训练模型，并在其基础上进行改进。改进模型分别将 ERNIE 的输出输入到 BiGRU、TextCNN 和 RCNN 中，然后在相同的测试集条件下对这些模型的效果进行评估。

测试结果表明，这些改进模型的效果都有所提升，其中 ERNIE-BiGRU 和 ERNIE-TextCNN 的效果提升差异不大，比基础模型提升了约 0.80%，ERNIE-RCNN 的效果提升最明显，Micro F1=88.46%，相对于基础模型提升了 1.85%。具体结果如图 4.8 所示。同时，对于模型训练集损失和验证集效果的观察发现，ERNIE-RCNN 的训练误差下降速度更快，而且最终收敛更加平稳。具体结果如图 4.9 所示。

进一步分析表明，BiGRU 可以对 ERNIE 学习到的语义信息进行进一步的学习，捕获更多上下文特征和学习更全面的信息。TextCNN 则能够捕获文本的全局语义信息和局部特征信息。RCNN 将 ERNIE 捕获到的信息输入到 BiGRU 中，然后将两者的输出拼接在一起，更好地汇聚信息。在解决长文本分类问题时，RCNN 效果更好。

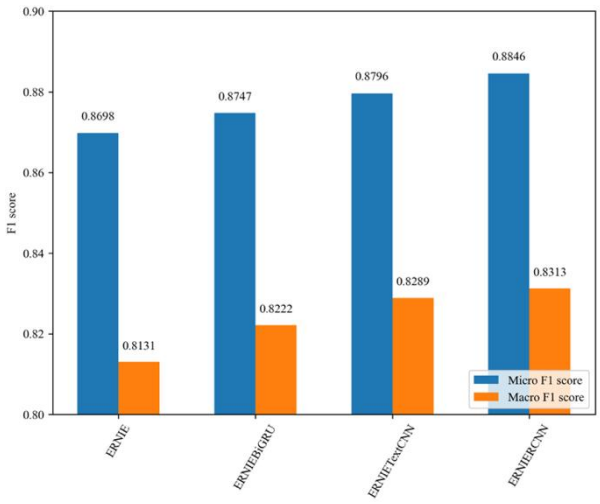
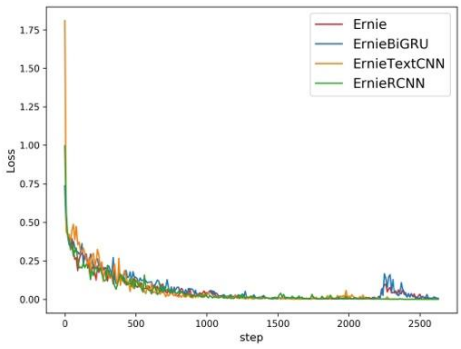
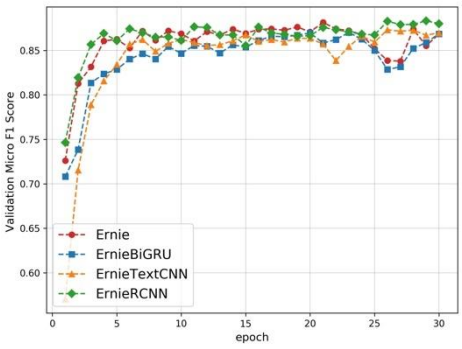


图 4.8 基于 ERNIE 的改进模型实验结果



（a）基于 ERNIE 的改进模型损失曲线



（b）基于 ERNIE 的改进模型验证曲线

图 4.9 基于 ERNIE 的改进模型结果曲线

4.5.3 基于数据增强的改进模型

如表 4.9 所示，在使用 TrustAI 进行稀疏数据筛选时，设置数据增强类型为 substitute，筛选稀疏数据数量为 60，用于数据增强的支持数据数量为 240，支持数据的阈值为 0.7。接下来，将筛选出的支持数据与原始训练集进行合并以实现数据增强。

数据增强的参数设置如表 4.10 所示，其中数据增强类型为随机替换（substitute），词替换增强类型为同义词方式（synonym），生成句子数量为当前训练集的 2 倍，生成词替换百分比为 0.1。在数据增强之后，训练集大小扩展至 4950 条对话，验证集大小保持为 200 条对话，测试集大小保持为 400 条对话。

根据表 4.11 的结果，经过数据增强的模型在测试集上的 Micro F1 达到了 88.99%，相较于数据增强前提升了 0.53%。同时，模型的 Macro F1 为 83.54%，相较于之前提升了 0.41%。这些结果表明，使用数据增强能够有效提升模型的性能。

表 4.9 稀疏数据筛选参数设置

参数名称	参数值
aug_strategy	substitute
sparse_num	60
support_num	240
support_threshold	0.7

表 4.10 数据增强参数设置

参数名称	参数值
aug_strategy	substitute
aug_type	synonym
create_n	2
support_threshold	0.1

表 4.11 数据增强模型对比

模型	Micro F1(%)	Macro F1(%)
ERNIE-RCNN	88.46	83.13
ERNIE-RCNN+数据增强	**88.99**	**83.54**

4.5.4 实验结果总结

最终经数据增强的 ERNIE-RCNN 模型相较于 BERT-base 模型 Micro F1 提升了 6.27%，较 ERNIE-base 提升了 1.98%，较未进行数据增强的 ERNIE-RCNN 提升了 0.53%。

各类标签具体表现如表 4.12 所示。分析具体标签的情况发现：流量、套餐、卡号、工单、用户信息共五个标签的效果较好，模型学习到了有关的信息，F1 score 均大于 90%。账单、业务、活动标签的效果较差，可能是这些标签对语义的要求更高，需要更深更大的预训练模型。机顶盒标签的效果较差，原因可能是机顶盒标签在数据标注时就较少，需要更多的有关数据进行训练。其他标签代表不属于上述十个标签的情况，也需要更多的有关数据进行训练。

表 4.12 各标签具体表现

标签	Precision(%)	Recall(%)	F1 Score(%)
账单	74.43	78.43	76.36
流量	88.42	94.92	91.55
套餐	93.46	96.62	95.01
卡号	89.08	92.17	90.60
业务	73.33	78.57	75.86
宽带	95.65	90.16	92.83
机顶盒	87.50	70.00	77.78
工单	87.94	92.88	90.35
活动	78.05	76.19	77.11
用户信息	91.07	90.27	90.67
其他	80.00	76.19	78.05

4.6 本章小结

本章针对第三章提出的基于预训练模型的对话领域多分类方法进行实验实现和验证。首先，介绍了本文所使用数据集的相关情况，包括标注准则、规模和数据分布。接下来，阐述了使用预训练模型解决多分类问题的实验流程，并通过实验及其结果分析，证实了 ERNIE-RCNN 模型的科学性和优越性。

5 总结与展望

5.1 工作总结

随着自然语言处理技术的发展，对话系统逐渐成为了人工智能应用的重要领域之一，并且其应用范围不断扩大，已经成为人们生活和工作中不可或缺的一部分。在对话系统中，自然语言理解模块是衡量其智能化程度的重要因素，作为其重要组成部分，对话领域多分类在其中发挥着重要的作用。然而，对话系统在中文领域仍面临着数据集匮乏、ASR 识别不准确、用户表述不规范以及对话多领域混合等问题的挑战。为了应对这些挑战，本文自主标注了中文对话多分类数据集，并提出了基于预训练模型的端到端领域多分类方法，对预训练基础模型进行了改进，以解决对话领域多分类的问题。

本文的主要研究内容及创新点如下：

（1）本文在公开数据集 MobileCS 的基础上筛选出了 2000 条高质量真实对话文本进行标注，之后又进行了数据增强对数据集进行扩充。最终得到的数据集共 5550 条对话数据，包括 11 个领域，每条数据都包含一个或多个对应的领域标签。

（2）本文提出了一种基于预训练模型的领域多分类方法，并进行了相应的算法实现。在此基础上，本文对 ERNIE 基础模型进行了优化与改进。通过对比实验和数据增强策略的应用，验证了本文所提出的 ERNIE-RCNN 模型在有效性和科学性方面的优越表现。

5.2 未来展望

本文提出了一种基于预训练模型的领域多分类方法，并在实验数据集上取得了令人满意的成果。尽管当前的模型性能已经初步满足任务导向型对话系统的需求，但对于自然语言处理任务而言，数据集规模相对较小。为了进一步提升模型性能，未来的工作可以考虑以下两个方向：

（1）模型轻量化：目前所训练的最终模型参数较多，整体模型较大，这不利于后续的模型部署。因此，可以进行模型轻量化的工作，例如应用知识蒸馏、模型剪枝和量化等方法。

（2）数据集扩充：当前所使用的数据集规模较小，而 MobileCS 数据集中尚有大量未标注数据。为提高模型性能，可以借助半监督方法扩充数据集，从而进一步提升模型效果。

致谢

大学四年，疫情三年，仿佛昨日依稀，时光荏苒如白驹过隙。转瞬间，我即将告别曲园。回顾曲园的求学岁月，我收获满满，纵有万般不舍，皆是感恩。

师恩之情，无以言报。首先我要向刘红娟老师致以衷心的感谢和敬意。大二的操作系统课程让我有幸结识了您，您对知识的深入理解、对学生的耐心教导和强烈的责任感深深感染了我。在推免过程中，您给予了我巨大的支持，我很幸运能够在您的指导下完成毕业论文设计。论文的顺利完成离不开您的悉心指导和耐心批改。同时，我要感谢我的研究生导师郑岩老师。从论文选题的确定、大纲的精心打磨到实验方案的制定与执行，都离不开您的辛勤付出。在我大学最后一年的学习和生活中，您给予了我极大的帮助，让我感受到了家人般的关爱与温暖。尽管言辞难以表达，但你们的师恩我将永志于心。

春晖寸草，山高海深。感谢我的父母一直以来默默无闻的付出，感谢父母的养育之恩与无限支持，他们始终关心我的成长，为我提供了一个充满爱与关怀的环境，是我永远的后盾。正因为他们的支持与鼓励，我得以坚定地踏上求知之路，勇敢迎接生活中的种种挑战。愿时光能够流转得更慢些，岁月赋予我们更多宝贵时光。养育之恩难以回报，愿我未来成为你们的骄傲与力量。

山水一程，三生有幸。感谢康一一同学，相识三载，相爱于今，即是知己，亦是伴侣。你的鼓励和欣赏，积极乐观的生活态度与善良品质，激发了我的创造力，助我战胜了科研道路和生活中的诸多困难。愿我们相守期明日，相知共成长，相伴迎未知；感谢远在重庆的兄弟李学顺、高欢，以及曲园的朋友辛沛霖、张子禹，还有 516 宿舍的小伙伴们，我们的相遇皆是宝贵缘分，感谢你们让我的本科岁月充满欢乐，谢谢你们在学习和生活中给予我的支持。愿你们生活顺遂、前程似锦，未来我们并肩前行。

最后，感谢对论文进行评审的老师与专家们，感谢你们在百忙之中抽出时间对我的论文进行指导。

至此，又是一个阶段的结束。在以后的人生道路上，我会谨记“学而不厌，诲人不倦”的校训，努力拼搏，不负期望！

参考文献

[1] TUR G, DE MORI R. Spoken language understanding: Systems for extracting semantic information from speech[M]. John Wiley & Sons, 2011.

[2] MORITZ N, HORI T, LE J. Streaming Automatic Speech Recognition with the Transformer Model[C/OL]//ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020: 6074-6078.

[3] WANG D, WANG X, LV S. An Overview of End-to-End Automatic Speech Recognition[J/OL]. Symmetry, 2019, 11(8): 1018.

[4] KIM S B, HAN K S, RIM H C, 等. Some Effective Techniques for Naive Bayes Text Classification[J/OL]. Knowledge and Data Engineering, IEEE Transactions on, 2006, 18: 1457-1466.

[5] MINAEE S, KALCHBRENNER N, CAMBRIA E, 等. Deep Learning Based Text Classification: A Comprehensive Review[M/OL]. arXiv, 2021[2023-03-17].

[6] COLAS F, BRAZDIL P. Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks[C/OL]//BRAMER M. Artificial Intelligence in Theory and Practice. Boston, MA: Springer US, 2006: 169-178.

[7] MIKOLOV T, CHEN K, CORRADO G, 等. Efficient Estimation of Word Representations in Vector Space[M/OL]. arXiv, 2013[2023-03-23].

[8] JOHNSON R, ZHANG T. Deep Pyramid Convolutional Neural Networks for Text Categorization[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 562-570[2023-03-29].

[9] TAI K S, SOCHER R, MANNING C D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks[M/OL]. arXiv, 2015[2023-03-29].

[10] LIU P, QIU X, CHEN X, 等. Multi-Timescale Long Short-Term Memory Neural Network for Modelling Sentences and Documents[C/OL]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 2326-2335[2023-03-29].

[11] LIU P, QIU X, HUANG X. Recurrent Neural Network for Text Classification with Multi-Task Learning[M/OL]. arXiv, 2016[2023-03-29].

[12] KIM Y. Convolutional Neural Networks for Sentence Classification[M/OL]. arXiv, 2014[2023-03-29].

[13] VASWANI A, SHAZEER N, PARMAR N, 等. Attention is All you Need[C/OL]//Advances in Neural Information Processing Systems: 卷 30. Curran Associates, Inc., 2017[2023-03-29].

[14] DEVLIN J, CHANG M W, LEE K, 等. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[M/OL]. arXiv, 2019[2023-03-29].

[15] RADFORD A, NARASIMHAN K, SALIMANS T, 等. Improving Language Understanding by Generative Pre-Training[J].

[16] OPENAI. GPT-4 Technical Report[M/OL]. arXiv, 2023[2023-03-31].

[17] BUBECK S, CHANDRASEKARAN V, ELDAN R, 等. Sparks of Artificial General Intelligence: Early experiments with GPT-4[M/OL]. arXiv, 2023[2023-03-31].

[18] TANG G, MÜLLER M, RIOS A, 等. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures[M/OL]. arXiv, 2018[2023-04-04].

[19] DENG J, DONG W, SOCHER R, 等. ImageNet: A large-scale hierarchical image database[C/OL]//2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009: 248-255.

[20] ILIĆ S, MARRESE-TAYLOR E, BALAZS J A, 等. Deep contextualized word representations for detecting sarcasm and irony[M/OL]. arXiv, 2018[2023-03-29].

[21] TSAI H, RIESA J, JOHNSON M, 等. Small and Practical BERT Models for Sequence Labeling[M/OL]. arXiv, 2019[2023-03-29].

[22] VAN AKEN B, WINTER B, LÖSER A, 等. How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations[C/OL]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. New York, NY, USA: Association for Computing Machinery, 2019: 1823-1832[2023-03-29].

[23] XU H, LIU B, SHU L, 等. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis[M/OL]. arXiv, 2019[2023-03-29].

[24] ADHIKARI A, RAM A, TANG R, 等. DocBERT: BERT for Document Classification[M/OL]. arXiv, 2019[2023-03-29].

[25] SUN Y, WANG S, LI Y, 等. ERNIE: Enhanced Representation through Knowledge Integration[M/OL]. arXiv, 2019[2023-03-19].

[26] HINTON G E. Learning distributed representations of concepts[C]//Proceedings of the eighth annual conference of the cognitive science society: 卷 1. Amherst, MA, 1986: 12.

[27] SUTSKEVER I, VINYALS O, LE Q V. Sequence to Sequence Learning with Neural Networks[C/OL]//Advances in Neural Information Processing Systems: 卷 27. Curran Associates, Inc., 2014[2023-04-01].

[28] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[M/OL]. arXiv, 2016[2023-04-02].

[29] HE K, ZHANG X, REN S, 等. Deep Residual Learning for Image Recognition[C/OL]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778[2023-03-29].

[30] LIU Y, OTT M, GOYAL N, 等. RoBERTa: A Robustly Optimized BERT Pretraining Approach[M/OL]. arXiv, 2019[2023-03-19].

[31] YANG Z, DAI Z, YANG Y, 等. XLNet: Generalized Autoregressive Pretraining for Language Understanding[C/OL]//Advances in Neural Information Processing Systems: 卷 32. Curran Associates, Inc., 2019[2023-04-04].

[32] DAI Z, YANG Z, YANG Y, 等. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context[M/OL]. arXiv, 2019[2023-04-04].

[33] SUN Y, WANG S, LI Y, 等. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding[J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(05): 8968-8975.

[34] SUN Y, WANG S, FENG S, 等. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation[M/OL]. arXiv, 2021[2023-03-29].